Appendix A

A.   Example XML Documents

Below are example instances of each document type:

B.   Init

The “init” document contains the name of the application and a human readable description.

```
<?xml version="1.0" ?>
<xamInit xmlns="http://www.e-speak.net/schema/XAM/init">
        <serviceName>Virtual File System:version 3.3</serviceName>
        <serviceInstanceIdentifier>IP15.144.65.2:pid=432</serviceInstanceIdentifier>
</xamInit>
```

C.   Request

The “request” document details what is required.

```
<?xml version="1.0"?>
<xamRequest xmlns="http://www.e-speak.net/schema/XAM/request" >

        <informationModel>typeCatalog</informationModel>
        <informationModel>configurationList</informationModel>
        <informationModel>systemVariables</informationModel>
</xamRequest>
```

D.   Details

```
The "details" document describes the current model state.
<?xml version="1.0"?>
<xamDetails xmlns="http://www.e-speak.net/schema/XAM/details">
        <systemVariables>
                        <!--   enable content is a boolean java object   -->
                <enable>true</enable>
                <serviceIdentifier>2nd Virtual File System:version 3.2</serviceIdentifier>
                <serviceInstanceIdentifier>2nd IP15.144.65.2</serviceInstanceIdentifier>
                <clockResolution>2nd 0.015</clockResolution>
                        <!--   minReportInterval content must be a long java object   -->

                <minReportInterval>55555</minReportInterval>
        </systemVariables>
        <typeCatalog>
                <transaction>
                                <!-- Order of elements within any container -->
                                <!--    is not garunteed -->
                        <typeIdentifier>Download_file</typeIdentifier>
                        <description> file downloaded from the service</description>

                        <variableSpec>
                                <name>2nd File_name</name>
                                <class>string</class>
                                <key>true</key>
                        </variableSpec>
                        <variableSpec>
                                <name>2nd size</name>
                                <key>false</key>
                                <class>string</class>
                                <unit>2nd bytes</unit>
                        </variableSpec>
                        <variableSpec>
                                <name>2nd size</name>
                                <key>false</key>
                                <class>string</class>
                                        <!--  unit is optional  -->
                                <unit>2nd bytes</unit>
                                        <!--  description is optional  -->
                                <description>sampling frame</description>
                        </variableSpec>
                        <variableSpec>
                                <name>File_name</name>
                                <!--   element "class" content must be "string" since  -->
                                <class>string</class>
                                <!--   key is optional, key content must be boolean    -->
                                <key>false</key>
                                <alphabet>
                                        <enum>1st apple</enum>
                                        <enum>1st orange</enum>
                                </alphabet>
                                <unit>transaction interval</unit>
                                <description>sampling frame</description>
                        </variableSpec>
                        <variableSpec>
                                <name>2nd aLongKeyVar</name>
                <!--  "class" content must be long since -->
                <!--   element range is used below       -->
            <class>long</class>
            <key>true</key>
            <alphabet>
                <range>2nd 0..100</range>
                <range>2nd 200..i</range>
            </alphabet>
            <unit>2nd percent</unit>
            <description>2nd (B.)  NO BUGS  </description>
        </variableSpec>
</transaction>
<count>
        <typeIdentifier>2nd testAtomic:count</typeIdentifier>
        <derivedTypeIdentifier>2nd testAtomic</derivedTypeIdentifier>

        <variableSpec>
            <name>2nd aLongKeyVar</name>
            <class>long</class>
            <key>true</key>
            <alphabet>
                <range>2nd 0..100</range>
                <range>2nd 200..i</range>
            </alphabet>
            <unit>2nd percent</unit>
            <description>size the file downloaded</description>
        </variableSpec>
        <variableSpec>
            <name>2nd aLongKeyVar</name>
            <class>long</class>
            <key>true</key>
            <alphabet>
                <range>2nd 0..100</range>
                <range>2nd 200..i</range>
            </alphabet>
            <unit>2nd percent</unit>
            <description>2nd (B.)  NO BUGS  </description>
        </variableSpec>
        <variableSpec>
            <name>2nd File_name</name>
            <class>string</class>
            <key>true</key>
        </variableSpec>

</count>

<sum>
        <variableSpec>
            <name>2nd File_name</name>
            <class>string</class>
            <key>true</key>
        </variableSpec>
        <variableSpec>
            <name>2nd aStringKeyVar</name>
            <class>string</class>
            <key>true</key>
            <alphabet>
                    <!-- Content must be unique -->
                <enum>apple1</enum>
                <enum>orange1</enum>
                <enum>apple2</enum>
                <enum>orange2</enum>
            </alphabet>
            <unit>2nd percent</unit>
            <description>2nd the user utilizing the disk</description>
        </variableSpec>
        <variableSpec>
            <name>2nd aStringVar</name>
            <class>string</class>
            <key>false</key>
            <alphabet>
                <enum>apple</enum>
                <enum>orange</enum>
            </alphabet>
            <unit>percent</unit>
            <description>the user utilizing the disk</description>
        </variableSpec>
        <variableSpec>
            <name>2nd aLongKeyVar</name>
            <class>long</class>
            <key>true</key>
            <alphabet>
```

```
                        <!-- Content must be unique -->
                 <range>0..100</range>
                 <range>200..i</range>
                 <range>400..450</range>
                 <range>2nd 200..i</range>
                 <range>0..101</range>
                 <range>2rd 200..i</range>
            </alphabet>
            <unit>2nd percent</unit>
            <description>2nd (B.)  No BUGS  </description>
        </variableSpec>
        <typeIdentifier>2nd testTransaction:sum:aLongVar</typeIdentifier>
        <derivedTypeIdentifier>2nd testTransaction</derivedTypeIdentifier>
        <derivedVariableName>2nd aLongVar</derivedVariableName>
    </sum>

    <atomic>
        <typeIdentifier>testAtomic</typeIdentifier>
        <description>a test atomic</description>
        <variableSpec>
            <name>aStringVar</name>
            <class>string</class>
            <key>false</key>
            <alphabet>
                <enum>apple</enum>
                <enum>orange</enum>
            </alphabet>
            <unit>percent</unit>
            <description>the user utilizing the disk</description>
        </variableSpec>
    </atomic>

    <poll>
        <typeIdentifier>testPolling</typeIdentifier>
        <description>a Polling test</description>
    </poll>
    <poll>
        <typeIdentifier>testPoll</typeIdentifier>
        <description>a bigger poll test </description>
        <variableSpec>
            <name>aStringVar</name>
            <class>string</class>
            <key>false</key>
            <alphabet>
                <enum>polled_apples</enum>
                <enum>polled_oranges</enum>
            </alphabet>
            <unit>percent</unit>
            <description>the user polling a service</description>
        </variableSpec>
    </poll>
    <!--    threshold not shown here:     -->
    <!--    group   not shown here: -->
</typeCatalog>
<configList xmlns="http://www.e-speak.net/schema/XAM/configure" >
    <sum>
        <handle>2888hfshgfh-sd/handle>
        <typeIdentifier>test sum:Duration</typeIdentifier>
        <aggregationInterval>20000.0</aggregationInterval>
    </sum>
    <sum>
        <handle>2888888888888</handle>
        <typeIdentifier>test sum:Duration</typeIdentifier>
        <aggregationInterval>20000.0</aggregationInterval>
    </sum>
    <sum>
        <handle>2999999999999</handle>
        <typeIdentifier>2nd sum </typeIdentifier>
        <aggregationInterval>200.0</aggregationInterval>
    </sum>
    <transaction>
        <handle>100</handle>
        <typeIdentifier>Tran_2</typeIdentifier>
          <restriction> TRUE </restriction>
    </transaction>
    <poll>
        <handle>2</handle>
        <typeIdentifier>Poll_2</typeIdentifier>
          <callbackInterval>2</callbackInterval>
    </poll>
    <atomic>
        <handle>22</handle>
```

```
                <typeIdentifier>Security_Authorization_failure</typeIdentifier>
                 <restriction> False </restriction>
            </atomic>
            <poll>
                <handle>3</handle>
                <typeIdentifier>Poll_3</typeIdentifier>
                 <callbackInterval>3</callbackInterval>
            </poll>
        </configList>
</xamdetails>
```

E   Configure

The “configure” document details the settings which the XAM library should use.

```
<?xml version="1.0"?>
<xamConfigure xmlns="http://www.e-speak.net/schema/XAM/configure">
        <systemVariables>
                <enable>true</enable>
                <minReportInterval>15</minReportInterval>
        </systemVariables>
        <configList>
                <transaction>
                        <handle>1</handle>
                </transaction>
                <transaction>
                        <handle>2</handle>
                </transaction>
                <transaction>
                        <handle>1</handle>
                                <!-- optional element -->
                        <typeIdentifier>DownLoad_file</typeIdentifier>
                </transaction>
                <transaction>
                        <handle>2</handle>
                        <typeIdentifier>DownLoad_file</typeIdentifier>
                </transaction>
                <transaction>
                        <handle>100</handle>
                        <typeIdentifier>Tran_2</typeIdentifier>
                        <restriction> TRUE </restriction>
                </transaction>
                <transaction>
                <handle>101</handle>
                        <typeIdentifier>Tran_2</typeIdentifier>
                                <!-- optional element -->
                  <restriction> TRUE </restriction>
                </transaction>
                <transaction>
                        <handle>102</handle>
                        <restriction> TRUE </restriction>
                </transaction>
                <atomic>
                        <handle>1</handle>
                                <!-- Optional Element -->
                        <typeIdentifier>Security_Authorization_failure</typeIdentifier>
                </atomic>
                <atomic>
                        <handle>1</handle>
                </atomic>
                <poll>
                        <handle>1</handle>
                                <!-- Optional Element -->
                        <typeIdentifier>Poll_1</typeIdentifier>
                                <!-- Optional Element -->
                        <callbackInterval>1</callbackInterval>
                                <!-- Optional Element -->
                        <restriction> False </restriction>
                </poll>
                <poll>
                        <handle>1</handle>
                </poll>
                <count>
                        <handle>3</handle>
                        <typeIdentifier>Download_file-count</typeIdentifier>
                                <!-- Must be a float   -->
                        <aggregationInterval>15.0</aggregationInterval>
                                <!-- Optional Element -->
                        <topNSize>10</topNSize>
                                <!-- Optional Element 0...n, Content MUST be unique -->
                        <restriction> TRUE </restriction>
                        <collapseDimension>File_type1</collapseDimension>
                        <collapseDimension>File_type2</collapseDimension>
                <collapseDimension>File_type3</collapseDimension>
                <collapseDimension>File_type4</collapseDimension>
                <collapseDimension>File_type5</collapseDimension>
                <collapseDimension>Invoker_ID1</collapseDimension>
                <collapseDimension>Invoker_ID2</collapseDimension>
                <restriction> FAIL </restriction>
                <collapseDimension>Invoker_ID3</collapseDimension>
                <collapseDimension>Invoker_ID4</collapseDimension>
        </count>
        <count>
                <handle>3</handle>
                <typeIdentifier>Count_2</typeIdentifier>
                <aggregationInterval>150.1</aggregationInterval>
        </count>
        <sum>
                <handle>4</handle>
                        <!-- Optional Element -->
                <typeIdentifier>SumType</typeIdentifier>
                        <!-- Must be a float   -->
                <aggregationInterval>20000.00001</aggregationInterval>
                        <!-- Optional Element -->
                <topNSize>9</topNSize>
                        <!-- Optional 0...n element(s), content MUST be unique-->
                <collapseDimension>Coll_Dim_1</collapseDimension>
                 <restriction> TRUE </restriction>
                <collapseDimension>Coll_Dim_2</collapseDimension>
                <restriction> FALSE </restriction>
                <collapseDimension>Coll_Dim_3</collapseDimension>
                <collapseDimension>Coll_Dim_4</collapseDimension>
        </sum>
        <sum>
                <handle>4</handle>
                <typeIdentifier>SumType_2</typeIdentifier>
                <aggregationInterval>200.0</aggregationInterval>
        </sum>
        <threshold>
                <handle>5</handle>
                        <!-- Optional Element -->
                <restriction> TRUE </restriction>
                        <!-- Optional Element -->
                <typeIdentifier>ThreshType</typeIdentifier>
                        <!-- Must be a float   -->
                <aggregationInterval>1.025</aggregationInterval>
                        <!-- Optional Element -->
                <topNSize>11</topNSize>
                        <!-- Optional 0..n element(s), content MUST be unique -->
                <collapseDimension>Coll_Dim_1</collapseDimension>
                <collapseDimension>Coll_Dim_2</collapseDimension>
                <collapseDimension>Coll_Dim_3</collapseDimension>
                        <!-- Must be a float   -->
                <thresholdValue>30.0</thresholdValue>
        </threshold>
        <threshold>
                <handle>50</handle>
                <aggregationInterval>10250.0</aggregationInterval>
                <thresholdValue>300.0</thresholdValue>
        </threshold>
    </configList>
</xamConfigure>
```

F   Report

The “report” document contains the measurement results. Several periods may be combined in one documents.

```
<?xml version="1.0"?>
<xamReport xmlns="http://www.e-speak.net/schema/XAM/report" >
        <timeBase>2000-04-18T17:43:30.140-08:00</timeBase>
        <reportInterval>2nd 1.313</reportInterval>
        <!-- a bundle should contain only one repetitive sibling element. -->
        <bundle>
                <handle>100</handle>
                <typeIdentifier>testPollData_100</typeIdentifier>
                <pollData>
                                <!-- Content in Name Elemant must be unique -->
                                <!--    OR Content in value element must be unique -->
                        <variableValue>
                                <name>aStringVar1</name>
                                <value>long in the tooth1</value>
                        </variableValue>
                </pollData>
                <pollData>
                <variableValue>
                        <name>aStringVar2</name>
                        <value>long in the tooth1</value>
                        </variableValue>
        </pollData>
        <pollData>
                <variableValue>
                        <name>aStringVar1</name>
                        <value>long in the tooth2</value>
                </variableValue>
        </pollData>
        <pollData>
                <variableValue>
                        <name>aStringVar2</name>
                        <value>long in the tooth2</value>
                </variableValue>
        </pollData>
        <pollData>
                <!-- other instance data here -->
        </pollData>
        <!-- other instance data here -->
</bundle>
<bundle>
        <handle>2nd 2</handle>
        <typeIdentifier>2nd testTransaction</typeIdentifier>
        <transactionData>
                        <!-- Element Content must be unique -->
                <instanceID>1</instanceID>
                        <!-- Optional Element -->
                <parentCorrelator>2nd null</parentCorrelator>
                <timeBase>2nd 2000-04-18T17:43:30.250-08:00</timeBase>
                <duration>2nd 0.11</duration>
                <status> 2nd The status is still very good today .,. </status>
                        <!-- Content in Name Elemant must be unique -->
                <variableValue>
                        <name>aStringVar1</name>
                        <value>long in the tooth string1</value>
                </variableValue>
                <variableValue>
                        <name>aStringVar1</name>
                        <value>2nd long in the tooth string2</value>
                </variableValue>
                <variableValue>
                        <name>aStringVar2</name>
                        <value>long in the tooth string1</value>
                </variableValue>
                <variableValue>
                        <name>aStringVar2</name>
                        <value>2nd long in the tooth string2</value>
                </variableValue>
                <variableValue>
                        <name>2nd aLongVar</name>
                        <value>2nd 82</value>
                </variableValue>
                <variableValue>
                        <name>2nd aLongKeyVar</name>
                        <value>2nd 83</value>
                </variableValue>
                <variableValue>
                        <name>2nd aDoubleVar</name>
                        <value>2nd 86.1459</value>
                </variableValue>
        </transactionData>
        <transactionData>
                <instanceID>2</instanceID>
                <timeBase>2000-04-18T17:43:30.250-08:00</timeBase>
                <duration>0.11</duration>
                <status>The status is still very good today .,. </status>
                <variableValue>
                        <name>aStringVar1</name>
                        <value>long in the tooth string1</value>
                </variableValue>
        </transactionData>
        <transactionData>
                <instanceID>3</instanceID>
                <timeBase>2nd 2000-04-18T17:43:30.250-08:00</timeBase>
                <duration>2nd 0.11</duration>
                <status> 2nd The status is still very good today .,. </status>
                <variableValue>
                        <name>aStringVar1</name>
                        <value>long in the tooth string1</value>
                </variableValue>
        </transactionData>
</bundle>
<bundle>
        <handle>2nd 1</handle>
        <typeIdentifier>2nd testAtomic</typeIdentifier>
        <atomicData>
                        <!-- elment is optional -->
                <parentCorrelator>2nd test r</parentCorrelator>
                <timeBase>2nd 2000-04-18T17:43:30.250-08:00</timeBase>
                <variableValue>
                        <name>2nd aStringVar</name>
                        <value>2nd long in the tooth</value>
                </variableValue>
                <variableValue>
                        <name>2nd aStringKeyVar</name>
                        <value>2nd long in the tooth with a key hole</value>
                </variableValue>
                <variableValue>
                        <name>2nd aLongVar</name>
                        <value>2nd 1111111122222222233333333</value>
                </variableValue>
                <variableValue>
                        <name>2nd aLongKeyVar</name>
                        <value>2nd 88884444422222</value>
                </variableValue>
                <variableValue>
                        <name>2nd aDoubleVar</name>
                        <value>2nd 87.1459</value>
                </variableValue>
        </atomicData>
        <atomicData>
                <timeBase>2nd 2000-04-18T17:43:30.250-08:00</timeBase>
                <variableValue>
                        <name>2nd aStringVar</name>
                        <value>2nd long in the tooth</value>
                </variableValue>
        </atomicData>
        <atomicData>
                <timeBase>2nd 2001-04-18T17:43:30.250-08:00</timeBase>
                <variableValue>
                <name>2nd aStringVar</name>
                        <value>2nd long in the tooth</value>
                </variableValue>
                <!-- other instance data here -->
        </atomicData>
        <!-- other instance data here -->
<bundle>
        <handle>6</handle>
        <typeIdentifier>testTransaction:sum:Duration</typeIdentifier>
        <sumData>
                <timeBase>2000-04-18T17:43:28.500-08:00</timeBase>
                <aggregationInterval>null</aggregationInterval>
                <Ex>2.0620000000000003</Ex>
                <Ex2>0.24450000000000002</Ex2>
                <N>18.0</N>
                <Max>0.203</Max>
                <Min>0.109</Min>
        </sumData>
        <sumData>
                <timeBase>2000-04-18T17:43:28.500-08:01</timeBase>
                <aggregationInterval>null</aggregationInterval>
                <Ex>2.0620000000000003</Ex>
                <Ex2>0.24450000000000002</Ex2>
                <N>18.0</N>
                <Max>0.203</Max>
                <Min>0.109</Min>
        </sumData>
</bundle>
<bundle>
                <!-- handle context must be unique -->
        <handle>3</handle>
        <typeIdentifier>testAtomic:count</typeIdentifier>
        <countData>
                <timeBase>2000-04-18T17:43:28.484-08:00</timeBase>
                <aggregationInterval>null</aggregationInterval>
                <count>18</count>
        </countData>
</bundle>
<bundle>
        <handle>4</handle>
        <typeIdentifier>testAtomic:count</typeIdentifier>
        <countData>
```

```
                <timeBase>2000-04-18T17:43:28.484-08:00</timeBase>
                <aggregationInterval>null</aggregationInterval>
                <count>18</count>
        </countData>
        <countData>
                <timeBase>2000-04-18T17:43:28.484-08:00</timeBase>
                <aggregationInterval>null</aggregationInterval>
                <!-- count content must be unique if everything else is equal -->
                <count>19</count>
        </countData>
</bundle>
<!-- not shown here: Count with TopN -->
<!-- not shown here: Group with or without TopN -->
<!-- not shown here: Sum with or without TopN -->
<!-- not shown here: Threshold with or without TopN -->

</xamReport>
```

G. Close

The "close" document contains a description of the closure reason or nothing. If either the XAM enabled application or the XAM service receives a close message they should reply with another close message and cease further communication. If either party has send a close message they should not send further close messages upon receipt of a close message.

```
<?xml version="1.0"?>
<xamClose xmlns="http://www.espeak.org/XAM/close">
</xamClose>
```

H. Info

The "info" document contains some information that is outside the scope of measurements. For example if an error occurs then an info message could be sent indicating what happened and what action has been taken. Implementations should be more tolerant of info messages containing unexpected element that they may be for other documents.

```
<?xml version="1.0"?>
<xamInfo xmlns="http://www.espeak.org/XAM/info">
        <badDocument>
                <documentName>xamConfig</documentName>
                <status>ignored</status>
                <reason>invalid xml</reason>
        </badDocument>
</xamInfo>
```

I. Schemas

TBC

J. Example Java usage

1. test.java

```
import net.espeak.management.xam.*;
import net.espeak.management.xam.clock.*;

/**
 * This is a example program which uses the XAM interface.
 */
public class test implements XAMCallbackInterface {


    /**
     * the entry point.
     * @param args command line arguments
     */
    public static void main(String[] args) {
        System.out.println("XAM test running");
        XAMTime time = XAMClock.getTime();
        System.out.println("The time is:" + time.getISO8601Format());

        test t = new test();

        t.theMain(args);
    }

    /**
```

```
     * the instance main
     * @param args command line arguments
     */
    void theMain(String[] args) {
        XAMSession s = XAMFactory.createSession("test application v1.0",
                    "15.144.69.251:pid=12312");

        // disk_utilization example
        XAMVariableSpec du[] = {
                          s.createDoubleVariablespec("Utilization", new String[] {"0..100"},
"percent", "the ammount of time the disk was busy"),
                          s.createLongVariableSpec("Disk", true, new String[] {"1..i"}, null,
"the disk being used"),
            s.createStringVariableSpec("User", true, null, null,
                          "the user utilizing the disk")
        };
        XAMPollType diskUtilization =
            s.newPollMeasurementType("Disk_utilization", du, this);

        diskUtilization.setDescription("the percentage of the time a disk was being utilized
by a user");

        // Download file example
        XAMVariableSpec df[] = {
            s.createStringVariablespec("File_name", false, "The name of the file being
downloaded"),
            s.createStringVariableSpec("File_type", true, "The type of the file being
downloaded"),
            s.createLongVariableSpec("Size", false, "the size of the file being downloaded"),
            s.createStringVariableSpec("Invoker_ID", true,
                          "the user utilizing the disk")
        };
        XAMTransactionType downloadFile =
            s.newTransactionType("Download_file", df);

        downloadFile.setDescription("a file download from the file server");

        // Security_Authorization_failure example
        XAMVariablespec saf[] = {
            s.createStringVariableSpec("Conversation_type", true, "The XMI conversation
name"),
            s.createStringVariableSpec("Document_type", true, "The XML document schema
address"),
            s.createStringVariableSpec("Invoker_ID", true, "the sender of the message"),
            s.createStringVariableSpec("Recipient_ID", true, "the reciver of the message"),
            s.createStringVariableSpec("Reason_String", true,
                          "the failure reason"),
        };
        XAMAtomicType securityAuthFail =
            s.newAtomicType("Security_Authorization_failure", saf);

        securityAuthFail.setDescription("a message from a remote client failed an
authorization check");

        // during run time
        // disk file download
        XAMTransactionInstance dwnldfileInstance = downloadFile.begin();

        dwnldfileInstance.setStringValue("File_name", "/user/web/index.html");
        dwnldfileInstance.setStringValue("File_type", "html");
        dwnldfileInstance.setLongValue("Size", 23432);
        dwnldfileInstance.setStringValue("Invoker_ID", "guest");

        // ... download the file here ....
        dwnldfileInstance.end();

        // Authorization failure
        XAMAtomicInstance safai = securityAuthFail.getInstance();

        safai.setStringValue("Conversation_type",
                             "http://www.espeak.net/conversations/fdsds.xml");
        safai.setStringValue("Document_type",
                             "http://www.espeak.net/documents/fsaasfdsds.xml");
        safai.setStringValue("Invoker_ID",
                             "PK:SHA-1:f34Dff23rd32tre5fFG445rFrr");
        safai.setStringValue("Recipient_ID",
                             "PK:SHA-1:DFsdsfHeRe85F4g55ggyh6h6v5b");
        safai.setStringValue("Reason_String",
                             "auth fail code 42 (the certificate was corrupt)");
        safai.report();
    }
    /**
     * The call back method is in this example implemented by the application object.
     *
     * @param measurementInstance
     */
    public void requestingMeasurement(XAMPollInstance measurementInstance) {
        double utilization[][] = {
             {
                34, 56, 45, 45, 56
            }, {
                45, 36, 35, 15, 76
            }, {
                39, 26, 32, 45, 56
            }
        };
        String user[] = {
            "tom", "dick", "harry"
        };
        long diskNum[] = {
            11, 12, 13, 14, 15
        };

        for (int u = 0; u < user.length; u++) {
            measurementInstance.setStringValue("User", user[u]);
            for (int d = 0; d < diskNum.length; d++) {
                measurementInstance.setLongValue("disk", diskNum[d]);
                measurementInstance.setDoubleValue("Utilization",
                        utilization[u][d]);

                // now post this measurement data.
                measurementInstance.postValueSet();
            }
        }
    }
}
```

2. *XAM Java API*

The following Java API is a first draft for the Java XAM application library.

3. *XAMFactory.java*

```
/*$Id: XAMFactory.java,v 1.1.2.4 2000/05/12 14:32:30 olima Exp $*/
/*
 *****************************************************************************
 *                                                                           *
 * (c) Copyright 2000 Hewlett-Packard Company                                *
 *                                                                           *
 * This program is free software; you can redistribute it and/or modify it  *
 * under the terms of the GNU Lesser General Public License as published by  *
 * the Free Software Foundation; either version 2.1 of the License, or       *
 * (at your option) any later version.                                       *
 *                                                                           *
 * This program is distributed in the hope that it will be useful, but       *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public   *
 * License for more details.                                                 *
 *                                                                           *
 * You should have received a copy of the GNU Lesser General Public License  *
 * along with this program; if not, write to the Free Software Foundation,   *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                             *
 *                                                                           *
 *****************************************************************************
 */

package net.espeak.management.xam;
import java.lang.reflect.*;


/**
 * This class is a factory for instances of XAMSession the factory may return
 * different types of XAMSession depending on the system propertie
 * "XAM.implementation.class". The default implementation returned
 * is <code>net.espeak.management.xam.noop.NoopSession<code>
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.4 $ $Date: 2000/05/12 14:32:30 $
 */
public class XAMFactory {

    /**
     * The last session object to be created. A session object is created when
     * one does not exist or for every new named session after the first named
     * session.
     */
    private static XAMSessionImpl latestSession;


    /**
     * indicates whether the session object in <code>latestSession</code> is named or not.
     */
    private static boolean sessionIsNamed = false;


    /**
     * The system property that holds the name of the implementation class.
     */
    public static final String XAM_IMPLENTATION_PROPERTY =
        "XAM.implementation.class";


    /**
     * The default noop implementation class.
     */
    public static final Class DEFAULT_XAM_IMPLEMENTATION_CLASS =
        net.espeak.management.xam.noop.NoopSession.class;


    /**
     * The actual implementation class.
     */
    public static final Class theXAMImplementation =
        loadOfXAMImplementation();


    /**
     * The default noop implementation class name.
     */
    public static final String DEFAULT_XAM_IMPLEMENTATION_NAME =
        DEFAULT_XAM_IMPLEMENTATION_CLASS.getName();


    /**
     * The default noop implementation class name.
     */
    public static final String XAM_LOADFAILEXIT_PROPERTY =
        "XAM.implementation.loadexit";


    /**
     * The default class load failure for the XAM implementation.
     * If true the library will call <code>System.exit(1);</code>
     */
    public static final String DEFAULT_XAM_LOADFAILEXIT = "false";


    /**
     * Create a XAMSession instance. This should generally be called
     * once per application instance.
     * @param ApplicationVersionName an identifier for the application
     * and it's version.
     * @param ApplicationInstanceName an identifier for the application
     * instance. No other application or service of with the same
     * ApplicationVersionName should have the same ApplicationInstanceName.
     * @throws ClassNotFoundException this exception may be thrown if the XAM
     * implementaion cannot be loaded.
     */
    public static synchronized XAMSession createSession(String applicationVersionName,
        String applicationInstanceName) {

        // check that the application has supplied
        // some application names with content
        if (applicationVersionName == null
            || applicationVersionName.equals("")) {
            throw new XAMException("The ApplicationVersionName cannot be empty");
        }
        if (applicationInstanceName == null
            || applicationInstanceName.equals("")) {
```

```
        throw new XAMException("The ApplicationInstanceName cannot be empty");
    }

    // check if we alrady have a session created but without any
    // application names or we don't have a session yet.
    if (sessionIsNamed == true || latestSession == null) {

        // we already had a named session so create a new one for the new names
        // or we don't have a session yet.
        latestSession = createSessionObject();
    }

    // remember that we have a named session
    sessionIsNamed = true;

    // now put the application names into the session.
    latestSession.setApplicationNames(applicationVersionName,
        applicationInstanceName);
    return latestSession;
}


/**
 * Create a XAMSession instance. This should generally be called
 * once per application instance. The Application instance will be
 * randomally generated.
 * @param ApplicationVersionName an identifier for the application
 * and it's version.
 * @throws ClassNotFoundException this exception may be thrown if the XAM
 * implementaion cannot be loaded.
 */
public static synchronized XAMSession createSession(String applicationVersionName) {

    // check that the application has supplied
    // some application names with content
    if (applicationVersionName == null
        || applicationVersionName.equals("")) {
        throw new XAMException("The ApplicationVersionName cannot be empty");
    }

    // check if we alrady have a session created but without any
    // application names or we don't have a session yet.
    if (sessionIsNamed == true || latestSession == null) {

        // we already had a named session so create a new one for the new names
        // or we don't have a session yet.
        latestSession = createSessionObject();
    }

    // remember that we have a named session
    sessionIsNamed = true;

    // now put the application names into the session.
    latestSession.setApplicationName(applicationVersionName);
    return latestSession;
}


/**
 * Create a XAMsession instance.
 * This method may be called by libraries that wish to use XAM but are not
 * part of the main application logic and cannot get to the <code>
 * XAMSession</code> instance returned by the full version of this method.
 * If the full version is called once then all calls to this method
 * (before or after the full version) will return the same <code>
 * XAMSession</code> instance. If mutiple calls are made to the full
 * version then calls to this method will return the latest <code>
 * XAMSession</code> instance return from a full call.
 * At least one call must be made to the full version or no measurements
 * will be recorded.
 */
public static synchronized XAMSession createSession()
    throws ClassNotFoundException {
    if (latestSession == null) {
        latestSession = createSessionObject();
    }
    return latestSession;
}


/**
 * create an instance of XAMSession.
 * The system property "XAM.implementation.class" is used
 * to get a class name which the method trys to load and
 * create and instance of.
 *
 * @return the newly created XAM session.
 */
private static XAMSessionImpl createSessionObject() {
    try {
        Class xamSessionClass = loadOfXAMImplementation();
        XAMSessionImpl theSession =
            (XAMSessionImpl)xamSessionClass.newInstance();

        // now check the the session can initialze itself
        // sucessfully
        theSession.initialze();

        return theSession;
    } catch (Exception e) {
        // keep trying not to throw any exceptions from this method
        return new net.espeak.management.xam.noop.NoopSession();
    }
}


/**
 * return the name of the implmentation class that we should load and use
 *
 * @return a fully qualified class name
 */
private static String implementationClassName() {
    String implClass = System.getProperty(XAM_IMPLENTATION_PROPERTY,
        DEFAULT_XAM_IMPLEMENTATION_NAME);

    return implClass;
}


/**
 * make sure the implementation is on the class path when this class is
 * loaded to avoid late failure due to the implementation not being on
 * the classpath. If class not found and the property
 * XAM_LOADFAILEXIT_PROPERTY == 'true' the program will be halted
 * with <code>System.exit(1);</code> otherwise the noop implementation
 * will be returned.
 */
private static Class loadOfXAMImplementation() {
    Class implementationClass = DEFAULT_XAM_IMPLEMENTATION_CLASS;
    try {
        // try to load the class in the system property.
        String implementationName = implementationClassName();

        if (implementationName == null) {
            // property not set so return noop class
            return DEFAULT_XAM_IMPLEMENTATION_CLASS;
        }

        try {
            // load the class
            implementationClass = Class.forName(implementationName);
        } catch(ClassNotFoundException e) {
            // failed to load the class
            if("true".equals(System.getProperty(XAM_LOADFAILEXIT_PROPERTY,
                DEFAULT_XAM_LOADFAILEXIT))) {
                // we should exit so rethrow exception
                throw e;
            } else {
                // just use the default.
                return DEFAULT_XAM_IMPLEMENTATION_CLASS;
            }
        }
        // check that the class actually subclasses the impl class
        if (XAMSession.class.isAssignableFrom(XAMSessionImpl.class) == false) {
            if ("true".equals(System.getProperty(XAM_LOADFAILEXIT_PROPERTY,
                DEFAULT_XAM_LOADFAILEXIT))) {
                throw new ClassNotFoundException("The XAM implemenetation class"
                    + " must extend:"
                    + net.espeak.management.xam.XAMSession.class.getName());
            } else {
                // just use the default.
                return DEFAULT_XAM_IMPLEMENTATION_CLASS;
            }
        }
        } catch (ClassNotFoundException e) {
            // a falure and the fail property set to false so exit.
            System.err.println("failed to load XAM implementation class: "
                + implementationClassName());
            System.err.println("XAM implementation class name is set in the system property
as: "
                + implementationClassName());
            System.err.println(e.toString());
            System.exit(1);
            return null;
        }
        return implementationClass;
    }
}
```

4. XAMSession java

```
/*$Id: XAMSession.java,v 1.1.2.3 2000/05/18 21:14:57 zs Exp $*/
/*
 *****************************************************************************
 *                                                                           *
 * (c) Copyright 2000 Hewlett-Packard Company                                *
 *                                                                           *
 * This program is free software; you can redistribute it and/or modify it   *
 * under the terms of the GNU Lesser General Public License as published by  *
 * the Free Software Foundation; either version 2.1 of the License, or       *
 * (at your option) any later version.                                       *
 *                                                                           *
 * This program is distributed in the hope that it will be useful, but       *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                 *
 *                                                                           *
 * You should have received a copy of the GNU Lesser General Public License  *
 * along with this program; if not, write to the Free Software Foundation,   *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                             *
 *                                                                           *
 *****************************************************************************
 */

package net.espeak.management.xam;


/**
 * This is the central type registration object from the applications
 * point of view. Calls are made on instances of this class inorder
 * to create measurement type instances.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.3 $ $Date: 2000/05/18 21:14:57 $
 */
public interface XAMSession {

    /**
     * create a new transaction type.
     * @param typeIdentifier a unique name for the transaction
     * measurement type.
     * @param variables an array of variable specifications
     * @param group this transaction type is to be included in
     * some XAMGroup's.
     * May be null or empty.
     */
    public abstract XAMTransactionType newTransactionType(String typeIdentifier,
        XAMVariableSpec variables[]) throws XAMException;


    /**
     * create a new atomic type.
     * @param typeIdentifier a unique name for the atomic measurement type.
     * @param variables an array of variable specifications
     * @param topN this atomic type is to be included in some XAMGroup's.
     * May be null or empty.
     */
    public abstract XAMAtomicType newAtomicType(String typeIdentifier,
        XAMVariableSpec variables[]) throws XAMException;


    /**
 * register a double callback object with the session. This will
 * cause the creation of a sample measurement.
 * @param valueName a unique name for this value.
 * @param variables an array of variable specifications
 * @param callbackObject an object which must implement the
 * XAMDoubleValue interface.
 */
public abstract XAMPollType newPollMeasurementType(String typeIdentifier,
    XAMVariableSpec variables[],
    XAMCallbackInterface callback) throws XAMException;


/**
 * create a variable specification which can be used when calling the
 * methods above.
 * @param name the name of the variable. must be unqiue within a
 * measurement type.
 * @param isKey set to true if the variable could be used to cut the
 * data but not be summed or thresholded.
 * @param Alphabet the scope of possible values for the variable.
 * E.g for  a string { "ready", "running", "stopped" }
 * or for a long or double a range spec { "11","12","15","32..126" }
 * @param unit A string describing the unit of the variable. E.g.
 * 'kilos', 'sessions', 'requests', 'seconds'
 * @param description Can contain any string, might contain a URL
 * to a localization service.
 */
public abstract XAMVariableSpec createStringVariableSpec(String name,
    boolean isKey, String[] alphabet, String unit,
    String description) throws XAMException;


/**
 * create a variable specification which can be used when calling the
 * methods above.
 * @param name the name of the variable. must be unqiue within a
 * measurement type.
 * @param isKey set to true if the variable could be used to cut the
 * data but not be summed or thresholded.
 * @param Alphabet the scope of possible values for the variable.
 * E.g for  a string { "ready", "running", "stopped" }
 * or for a long or double a range spec { "11","12","15","32..126" }
 * @param unit A string describing the unit of the variable. E.g.
 * 'kilos', 'sessions', 'requests', 'seconds'
 * @param description Can contain any string, might contain a URL
 * to a localization service.
 */
public abstract XAMVariableSpec createLongVariableSpec(String name,
    boolean isKey, String[] alphabet, String unit,
    String description) throws XAMException;


/**
 * create a variable specification which can be used when calling the
 * methods above.
 * @param name the name of the variable. must be unqiue within a
 * measurement type.
 * @param isKey set to true if the variable could be used to cut the
 * data but not be summed or thresholded.
 * @param Alphabet the scope of possible values for the variable.
 * E.g for  a string { "ready", "running", "stopped" }
 * or for a long or double a range spec { "11","12","15","32..126" }
 * @param unit A string describing the unit of the variable. E.g.
 * 'kilos', 'sessions', 'requests', 'seconds'
 * @param description Can contain any string, might contain a URL
 * to a localization service.
 */
public abstract XAMVariableSpec createDoubleVariableSpec(String name,
    String[] alphabet, String unit,
    String description) throws XAMException;


/**
 * create a long variable specification which can be used when calling the
 * methods above.
 * @param name the name of the variable. must be unqiue within a
 * measurement type.
 * @param className one of "string", "long" or "double".
 * @param isKey set to true if the variable could be used to cut the data
 * but not be summed or thresholded.
 * @param description Can contain any string, might contain a URL to a
 * localization service.
```

```
     */
    public XAMVariableSpec createLongVariableSpec(String name, boolean isKey,
        String description) throws XAMException;

    /**
     * create a string variable specification which can be used when calling the
     * methods above.
     * @param name the name of the variable. must be unqiue within a
     * measurement type.
     * @param className one of "string", "long" or "double".
     * @param isKey set to true if the variable could be used to cut the data
     * but not be summed or thresholded.
     * @param description can contain any string, might contain a URL to a
     * localization service.
     */
    public XAMVariableSpec createStringVariableSpec(String name,
        boolean isKey, String description) throws XAMException;

    /**
     * create a double variable specification which can be used when calling the
     * methods above.
     * @param name the name of the variable. must be unqiue within a
     * measurement type.
     * @param className one of "string", "long" or "double".
     * @param isKey set to true if the variable could be used to cut the data
     * but not be summed or thresholded.
     * @param description can contain any string, might contain a URL to a
     * localization service.
     */
    public XAMVariableSpec createDoubleVariableSpec(String name,
        String description) throws XAMException ;

    /**
     * Can be used by the application to check if the <code>XAMSession</code>
     * implemenetation have recvied configuration information from the XAM
     * service. An application may wait until this value is true inorder not
     * to loose any measurement data. Applications should not wait idenfiatly
     * for a true value as the XAM library may not find a XAM service.
     */
    public boolean configured();
}
```

5. *XAMMeasurementType.java*

```
/*$Id: XAMMeasurementType.java,v 1.1.2.2 2000/05/08 01:04:13 ds Exp $*/
/*
 ***************************************************************************************
 *                                                                                      *
 * (c) Copyright 2000 Hewlett-Packard Company                                          *
 *                                                                                      *
 * This program is free software; you can redistribute it and/or modify it             *
 * under the terms of the GNU Lesser General Public License as published by            *
 * the Free Software Foundation; either version 2.1 of the License, or                 *
 * (at your option) any later version.                                                  *
 *                                                                                      *
 * This program is distributed in the hope that it will be useful, but                 *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY           *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public              *
 * License for more details.                                                            *
 *                                                                                      *
 * You should have received a copy of the GNU Lesser General Public License            *
 * along with this program; if not, write to the Free Software Foundation,             *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                                        *
 *                                                                                      *
 ***************************************************************************************
 */

package net.espeak.management.xam;


/**
 * The top level class for Measurement type hierarchy.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.2 $ $Date: 2000/05/08 01:04:13 $
 */
public interface XAMMeasurementType {
    /**
     * set the description of the measurement type
     */
    public void setDescription(String description);

    /**
     * return the String form of the correlator for this transaction type.
     * The correlator is a unicode string of the
     * form "System Identifier!System Instance Identifier!Type
     * Identifier!"
     * May be passed by the application to
     * component application which could use it to correlate two Occurrences
     *
     * @return String form of correlator.
     */
    public String getCorrelatorAsString();
}
```

6. XAMOccuranceType.java

```
/*$Id: XAMOccurrenceType.java,v 1.1.2.1 2000/05/08 01:04:16 ds Exp $*/
/*
 ******************************************************************************
 *                                                                            *
 * (c) Copyright 2000 Hewlett-Packard Company                                 *
 *                                                                            *
 * This program is free software; you can redistribute it and/or modify it   *
 * under the terms of the GNU Lesser General Public License as published by   *
 * the Free Software Foundation; either version 2.1 of the License, or        *
 * (at your option) any later version.                                        *
 *                                                                            *
 * This program is distributed in the hope that it will be useful, but        *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                  *
 *                                                                            *
 * You should have received a copy of the GNU Lesser General Public License   *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
 ******************************************************************************
 */

package net.espeak.management.xam;

/**
 * This is an abstract interface no actual instances
 * of this inteface should exist. The two concrete subclasses of
 * are <code>XAMAtomicType</code> and <code>XAMTransactionType</code>
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.1 $ $Date: 2000/05/08 01:04:16 $
 */
public interface XAMOccurrenceType extends XAMMeasurementType {

    /**
     * If this measurement type is not currently required by the library
     * then the application could save time by not creating the value set.
     * This value may change at any time so MUST be checked on a case by
     * case basis.
     * Do not cache the return value from this method!!
     * @return indication if the measurement type is being used by the library
     */
    public boolean enabled();
}
```

7. XAMAtomicType.java

```
/*$Id: XAMAtomicType.java,v 1.1.2.2 2000/05/08 01:04:10 ds Exp $*/
/*
 ******************************************************************************
 *                                                                            *
 * (c) Copyright 2000 Hewlett-Packard Company                                 *
 *                                                                            *
 * This program is free software; you can redistribute it and/or modify it   *
 * under the terms of the GNU Lesser General Public License as published by   *
 * the Free Software Foundation; either version 2.1 of the License, or        *
 * (at your option) any later version.                                        *
 *                                                                            *
 * This program is distributed in the hope that it will be useful, but       *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                  *
 *                                                                            *
 * You should have received a copy of the GNU Lesser General Public License  *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
******************************************************************************
 */

package net.espeak.management.xam;


/**
 * One instance of this class should exist for every Atomic measurement
 * type registered by the application. Each time the application
 * wishes to record an instance of this measurement type the
 * <code>getInstance</code> method must be called to create
 * a new instance of a measurement.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.2 $ $Date: 2000/05/08 01:04:10 $
 */
public interface XAMAtomicType extends XAMOccurrenceType {


    /**
     * Indicate that the Atomic Occurrence has happened.Each time the
     * application wishes to record an instance of this measurement
     * type the this method must be called to create a new instance
     * of a measurement.
     *
     * @return atomic measurement instance
     */
    public XAMAtomicInstance getInstance();
}
```

8. XAMTransactionType.java

```
/*$Id: XAMTransactionType.java,v 1.1.2.2 2000/05/08 01:04:18 ds Exp $*/
/*
******************************************************************************
 *                                                                            *
 * (c) Copyright 2000 Hewlett-Packard Company                                 *
 *                                                                            *
 * This program is free software; you can redistribute it and/or modify it   *
 * under the terms of the GNU Lesser General Public License as published by   *
 * the Free Software Foundation; either version 2.1 of the License, or        *
 * (at your option) any later version.                                        *
 *                                                                            *
 * This program is distributed in the hope that it will be useful, but       *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                  *
 *                                                                            *
 * You should have received a copy of the GNU Lesser General Public License  *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
******************************************************************************
 */

package net.espeak.management.xam;
import net.espeak.management.xam.clock.XAMTime;


/**
 * One instance of this class should exist for every Transaction measurement
 * registered by the application. Each time the application
 * wishes to record an instance of this measurement type one of the
 * <code>begin()</code> methods must be called to create
 * a new instance of a transaction measurement.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.2 $ $Date: 2000/05/08 01:04:18 $
 */
public interface XAMTransactionType extends XAMOccurrenceType {

    /**
     * Inform XAM that a new transaction has just started.
     * @return a new instance of transaction measurement.
     */
    public XAMTransactionInstance begin();

    /**
     * Inform XAM about a new transaction that started some time ago.
     * @param time the time instant the transaction started.
     * @return a new instance of transaction measurement.
     */
    public XAMTransactionInstance began(XAMTime time) throws XAMException;

    /**
     * Inform XAM about a new transaction that started some time ago.
     * @param time the time instant the transaction started. The value should
     * be in ISO 8601 format. Only time values generated from XAM
     * libraries should be used here.
     * @return a new instance of transaction measurement.
     */
    public XAMTransactionInstance began(String time) throws XAMException;
}
```

9.   XAMOccuranceInstance.java

```
/*$Id: XAMOccurrenceInstance.java,v 1.1.2.1 2000/05/08 01:04:15 ds Exp $*/
/*
 ******************************************************************************
 *                                                                            *
 * (c) Copyright 2000 Hewlett-Packard Company                                 *
 *                                                                            *
 * This program is free software; you can redistribute it and/or modify it    *
 * under the terms of the GNU Lesser General Public License as published by   *
 * the Free Software Foundation; either version 2.1 of the License, or        *
 * (at your option) any later version.                                        *
 *                                                                            *
 * This program is distributed in the hope that it will be useful, but        *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                  *
 *                                                                            *
 * You should have received a copy of the GNU Lesser General Public License   *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
 ******************************************************************************
 */

package net.espeak.management.xam;

/**
 * This is an abstract interface no actual instances
 * of this inteface should exist. The two concrete subclasses of
 * are <code>XAMAtomicInstance</code> and <code>XAMTransactionInstance</code>
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.1 $ $Date: 2000/05/08 01:04:15 $
 */
public interface XAMOccurrenceInstance extends XAMMeasurementInstance {

    /**
     * set the parent instance correlator. The correlator is a combination of
     * System Identifier, System Instance Identifier, Type
     * Identifier and Type Instance Identifier.
     * @param parentInstance the instance identifier of the parent.
     */
    public void setParentCorrelator(XAMOccurrenceInstance parentInstance)
```

```
        throws XAMException;

    /**
     * set the parent instance correlator. The correlator is a unicode string
     * of the form "System Identifier!System Instance Identifier!Type
     * Identifier!Type Instance Identifier"
     * @param parentInstance the instance identifier of the parent in String
     * form.
     */
    public void setParentCorrelator(String parentCorrelator)
        throws XAMException;
}
```

10. XAMTransactionInstance.java

```
/*$Id: XAMTransactionInstance.java,v 1.1.2.2 2000/05/08 01:04:17 ds Exp $*/
/*
 ******************************************************************************
 *                                                                            *
 * (c) Copyright 2000 Hewlett-Packard Company                                 *
 *                                                                            *
 * This program is free software; you can redistribute it and/or modify it   *
 * under the terms of the GNU Lesser General Public License as published by   *
 * the Free Software Foundation; either version 2.1 of the License, or        *
 * (at your option) any later version.                                        *
 *                                                                            *
 * This program is distributed in the hope that it will be useful, but        *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                  *
 *                                                                            *
 * You should have received a copy of the GNU Lesser General Public License   *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
 ******************************************************************************
 */

package net.espeak.management.xam;
import net.espeak.management.xam.clock.XAMTime;


/**
 * An instance of a Transaction measurement. Instances are created from
 * a XAMTransactionType object. Instances of this class can only be used
 * to report one measurement. Each time a new transaction measurement
 * occures a new instance of this class must be created.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.2 $ $Date: 2000/05/08 01:04:17 $
 */
public interface XAMTransactionInstance extends XAMOccurrenceInstance {


    /**
     * Indicate that the transactions has finished sucessfully the completed field
     * will be set to 'OK'. Measurement data will be generated for this instance.
     * All the variables that are defined for this measurement type need to be set
     * before calling this method.
     */
    public void end();


    /**
     * Indicate that the transactions has finished abnormally the completed field
     * will be set to 'FAIL'. Measurement data will be generated for this instance.
     * All the variables that are defined for this measurement type need to be set
     * before calling this method.
     */
    public void abort();


    /**
     * indicate that the transaction should be ignored for measurement
     * purposes. This might be used in cases such as transaction rollback
     * or transaction failure. where a different measurement type should
     * now be used. Measurement data will NOT be generated for this instance.
     */
    public void remove();
    /**
     * the time that the transaction started
     * @return the start time
     */
    public XAMTime startTime();

    /**
     * return the String form of the correlator for this transaction instance.
     * The correlator is a unicode string of the
     * form 'System Identifier!System Instance Identifier!Type
     * Identifier!Type Instance Identifier'
     * May be passed by the application to
     * component application which could use it to correlate two occurrences
     *
     * @return String form of correlator.
     */
    public String getCorrelatorAsString();
}
```

11. XAMCallbackInterface.java

```
/*$Id: XAMCallbackInterface.java,v 1.1.2.3 2000/05/18 21:14:56 zs Exp $*/
/*
 *****************************************************************************
 *                                                                           *
 * (c) Copyright 2000 Hewlett-Packard Company                                *
 *                                                                           *
 * This program is free software; you can redistribute it and/or modify it   *
 * under the terms of the GNU Lesser General Public License as published by  *
 * the Free Software Foundation; either version 2.1 of the License, or       *
 * (at your option) any later version.                                       *
 *                                                                           *
 * This program is distributed in the hope that it will be useful, but       *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public   *
 * License for more details.                                                 *
 *                                                                           *
 * You should have received a copy of the GNU Lesser General Public License  *
 * along with this program; if not, write to the Free Software Foundation,   *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                             *
 *                                                                           *
 *****************************************************************************
 */

package net.espeak.management.xam;

/**
 * This is the Poll measurement callback inteface. Some application
 * class must implement this inteface if the application registers Poll
 * measurements.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.3 $ $Date: 2000/05/18 21:14:56 $
 */
public interface XAMCallbackInterface {

    /**
     * This method is called by the XAM library every time it wishes to sample
     * the measurement. The application must make calls on the <code>
     * measurementInstance</code> object to post the measurement data.
     * References to the measurementInstance object should not be kept after
     * the method returns. Runtime exceptions thrown from implementations of this
     * method will be treated as a return.
     *
     * @see XAMPollInstance
     * @param measurementInstance
     */
    public void requestingMeasurement(XAMPollInstance measurementInstance);
}
```

12. XAMAtomicInstance.java

```
/*$Id: XAMAtomicInstance.java,v 1.1.2.2 2000/05/08 01:04:09 ds Exp $*/
/*
 *****************************************************************************
 *                                                                          *
 * (c) Copyright 2000 Hewlett-Packard Company                               *
 *                                                                          *
 * This program is free software; you can redistribute it and/or modify it *
 * under the terms of the GNU Lesser General Public License as published by *
 * the Free Software Foundation; either version 2.1 of the License, or      *
 * (at your option) any later version.                                      *
 *                                                                          *
 * This program is distributed in the hope that it will be useful, but      *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public  *
 * License for more details.                                                *
 *                                                                          *
 * You should have received a copy of the GNU Lesser General Public License *
 * along with this program; if not, write to the Free Software Foundation,  *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                            *
 *                                                                          *
****************************************************************************
 */

package net.espeak.management.xam;


/**
 * An instance of an Atomic measurement. Instances are created from
 * a XAMAtomicType object. Instances of this class can only be used
 * to report one measurement. Each time a new atomic measurement
 * occures a new instance of this class must be created.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.2 $ $Date: 2000/05/08 01:04:09 $
 */
public interface XAMAtomicInstance extends XAMOccurrenceInstance {


    /**
     * the value set data has been inserted, so the measurement library
     * can take the data. This method should be called once only per
     * object instance. If the method is called twice then an exception
     * will be thrown.
     */
    public void report();
}
```

13. XAMPollingType.java

```
/*$Id: XAMPollType.java,v 1.1.2.1 2000/05/18 21:19:18 zs Exp $*/
/*
****************************************************************************
 *                                                                          *
 * (c) Copyright 2000 Hewlett-Packard Company                               *
 *                                                                          *
 * This program is free software; you can redistribute it and/or modify it *
 * under the terms of the GNU Lesser General Public License as published by *
 * the Free Software Foundation; either version 2.1 of the License, or      *
 * (at your option) any later version.                                      *
 *                                                                          *
 * This program is distributed in the hope that it will be useful, but      *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public  *
 * License for more details.                                                *
 *                                                                          *
 * You should have received a copy of the GNU Lesser General Public License *
 * along with this program; if not, write to the Free Software Foundation,  *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                            *
 *                                                                          *
****************************************************************************
 */

package net.espeak.management.xam;


/**
 * One instance of this class should exist for every Poll measurement
 * registered by the application.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.1 $ $Date: 2000/05/18 21:19:18 $
 */
```

```
public interface XAMPollType extends XAMMeasurementType {

    /**
     * Return the name of the type which was registered by the application.
     * @returns the name the application gave to this Poll type.
     */
    public String getTypeIdentifier();
}
```

14. XAMException java

```
/*$Id: XAMException.java,v 1.1.2.2 2000/05/08 01:04:11 ds Exp $*/
/*
*****************************************************************************
 *                                                                          *
 * (c) Copyright 2000 Hewlett-Packard Company                               *
 *                                                                          *
 * This program is free software; you can redistribute it and/or modify it  *
 * under the terms of the GNU Lesser General Public License as published by *
 * the Free Software Foundation; either version 2.1 of the License, or      *
 * (at your option) any later version.                                      *
 *                                                                          *
 * This program is distributed in the hope that it will be useful, but      *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public  *
 * License for more details.                                                *
 *                                                                          *
 * You should have received a copy of the GNU Lesser General Public License *
 * along with this program; if not, write to the Free Software Foundation,  *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                            *
 *                                                                          *
*****************************************************************************
 */

package net.espeak.management.xam;


/**
 * The XAM library passes all runtime errors as instatnces of this exception
 * type.
 * The XAM library will only throw exception is the case of bad usage by the
 * application.
 * If the XAM library sufferers an internal error or cannot reach the XAM
 * service no exceptions will be thrown. If the application is programmed
 * correctly this exception will not be thrown.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.2 $ $Date: 2000/05/08 01:04:11 $
 */
public class XAMException extends RuntimeException {

    /**
     * Create a XAM Exception
     *
     * @param msg
     */
    public XAMException(String msg) {
        super(msg);
    }
}
```

15. XAMClock.java

```
}
/*$Id: XAMClock.java,v 1.1.2.5 2000/05/18 21:14:59 zs Exp $*/
/*
*****************************************************************************
 *                                                                          *
 * (c) Copyright 2000 Hewlett-Packard Company                               *
 *                                                                          *
 * This program is free software; you can redistribute it and/or modify it  *
 * under the terms of the GNU Lesser General Public License as published by *
 * the Free Software Foundation; either version 2.1 of the License, or      *
 * (at your option) any later version.                                      *
 *                                                                          *
 * This program is distributed in the hope that it will be useful, but      *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public  *
 * License for more details.                                                *
 *                                                                          *
 * You should have received a copy of the GNU Lesser General Public License   *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
 ******************************************************************************
 */

package net.espeak.management.xam.clock;
import java.lang.reflect.*;
import java.text.SimpleDateFormat;
import net.espeak.management.xam.XAMException;


/**
 * This is class gives access to the most accurate time available.
 * The clock will use an implementation class as per the system property
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.5 $ $Date: 2000/05/18 21:14:59 $
 */
public class XAMClock {
    /**
     * the time now.
     * @return the current time.
     */
    public static XAMTime getTime();


    /**
     * the clock resolution in seconds. i.e the smallest measurable
     * difference in time that the clock can produce.
     * @return the clock resolution time in seconds.
     */
    public static double getClockResolution();


    /**
     * create a XAMTime object from a string in ISO 8601 format.
     * ISO (International Organization for Standardization). Representations
     * of dates and times, 1988-06-15.
     * Available at: http://www.iso.ch/markete/8601.pdf
     * @return a string containing a time instant in the form
     * "CCYY-MM-DDThh:mm:ss.ssssss" or with time zone offset
     * eg "CCYY-MM-DDThh:mm:ss.ssssss+hh:mm"
     */
    public static XAMTime getTimeFromString(String time);


    /**
     * create an instance of XAMSession.
     * The system property "XAM.implementation.class" is used
     * to get a class name which the method trys to load and
     * create and instance of.
     *
     * @return the newly created XAM session.
     */
    private static XAMClockImpl createClock();


    /**
     * check that the time string is in ISO 8601 format.
     * ISO (International Organization for Standardization). Representations
     * of dates and times, 1988-06-15.
     * Available at: http://www.iso.ch/markete/8601.pdf
     * @return a string containing a time instant in the form
     * "CCYY-MM-ddThh:mm:ss.ssssss" or with time zone offset
     * "CCYY-MM-ddThh:mm:ss.ssssss+hh:mm" or "CCYY-MM-ddThh:mm:ss.ssssss-hh:mm"
     */
    public static void checkTime(String time);
}
```

16.   XAMTime.java

```
/*$Id: XAMTime.java,v 1.1.2.4 2000/05/18 21:14:59 zs Exp $*/
/*
 ******************************************************************************
 *                                                                            *
 * (c) Copyright 2000 Hewlett-Packard Company                                 *
 *
 * This program is free software; you can redistribute it and/or modify it    *
 * under the terms of the GNU Lesser General Public License as published by   *
 * the Free Software Foundation; either version 2.1 of the License, or        *
 * (at your option) any later version.                                        *
 *                                                                            *
 * This program is distributed in the hope that it will be useful, but        *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                  *
 *                                                                            *
 * You should have received a copy of the GNU Lesser General Public License   *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
 ******************************************************************************
 */

package net.espeak.management.xam.clock;


/**
 * This type is used by XAM to hold a time instant. There are no methods to
 * convert to/from java time formats. This is to discourage pollution of the
 * XAM time system from the java time system which may be different.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.4 $ $Date: 2000/05/18 21:14:59 $
 */
public interface XAMTime {


    /**
     * return the time in ISO 8601 format.
     * ISO (International Organization for Standardization). Representations
     * of dates and times, 1988-06-15.
     * Available at: http://www.iso.ch/markete/8601.pdf
     * @return a string containing a time instant in the form
     * "CCYY-MM-ddThh:mm:ss.ssssss" or with time zone offset
     * "CCYY-MM-ddThh:mm:ss.ssssss+hh:mm" or "CCYY-MM-ddThh:mm:ss.ssssss-hh:mm"
     */
    String getISO8601Format();


    /**
     * calculate the difference between the two times in seconds
     *
     * @param time the other time value
     */
    public double minus(XAMTime time);


    /**
     * Generate a new time in the future
     *
     * @param time in seconds to be added to this time
     */
    public XAMTime plus(double time);
}
```

17. XAMMeasurementInstance.java

```
/*$Id: XAMMeasurementInstance.java,v 1.1.2.1 2000/05/08 01:04:13 ds Exp $*/
/*
 ******************************************************************************
 *                                                                            *
 * (c) Copyright 2000 Hewlett-Packard Company                                 *
 *                                                                            *
 * This program is free software; you can redistribute it and/or modify it    *
 * under the terms of the GNU Lesser General Public License as published by   *
 * the Free Software Foundation; either version 2.1 of the License, or        *
 * (at your option) any later version.                                        *
 *                                                                            *
 * This program is distributed in the hope that it will be useful, but        *
 * WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY *
 * or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public    *
 * License for more details.                                                  *
 *                                                                            *
 * You should have received a copy of the GNU Lesser General Public License   *
 * along with this program; if not, write to the Free Software Foundation,    *
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.                              *
 *                                                                            *
 ******************************************************************************
 */

package net.espeak.management.xam;

/**
 * Through this interface several variable values may be set by the application.
 * The correct method must be called depending on the class type specified in the
 * variable declaration. A value may be writen more than once if necessary.
 *
 * @author David Stephenson
 *
 * @version $Revision: 1.1.2.1 $ $Date: 2000/05/08 01:04:13 $
 */
public interface XAMMeasurementInstance {

    /**
     * set a value of a variable.
     * @param name name of the variable.
     * @param value the value of the variable.
     */
    public void setStringValue(String name, String value) throws XAMException;

    /**
     * set a value of a variable.
     * @param name name of the variable.
     * @param value the value of the variable.
     */
    public void setLongValue(String name, long value) throws XAMException;

    /**
     * set a value of a variable.
     * @param name name of the variable.
     * @param value the value of the variable.
     */
    public void setDoubleValue(String name, double value) throws XAMException;
}
```